Improving the Diagnostic Performance for Dynamic Systems by Using Conflict-Driven Model Decomposition*

Anibal Bregon[1], Alexander Feldman[2], Belarmino Pulido[1], Gregory Provan[2], Carlos Alonso-González[1]
[1] Depto. de Informática, Universidad de Valladolid, Spain
[2] Dept. of Computer Science, University College Cork, Ireland

Abstract

This work studies potential ways of integration of two techniques for fault detection, isolation, and identification in dynamic systems: the LYDIA-NG suite of diagnosis algorithms and the Consistency-based Diagnosis approach with Possible Conflicts. By integrating both techniques, LYDIA-NG will benefit from a more efficient fault detection and isolation task, and Possible Conflicts will benefit from the identification capabilities of LYDIA-NG. In this paper, we define a common framework that integrates both techniques, and then we apply the proposed integrated approach to a three-tank system, and draw some conclusions about potential ways of integration.

1 Introduction

The need for safety and reliability in engineering systems provides the motivation for developing Integrated Systems Health Management (ISHM) methodologies that include efficient fault diagnosis mechanisms. In this work we focus on model-based approaches to on-line fault diagnosis of dynamic systems. Online methods for model-based diagnosis require the use of quick and robust fault detection methods to establish discrepancies between observed and expected system behavior. However, accurate and timely online fault diagnosis of complex dynamic systems is difficult and can be computationally expensive [Isermann, 2006].

In this work we study how to combine two techniques suitable for model-based diagnosis of dynamic systems looking for better performance in on-line fault diagnosis. We have used the LYDIA-NG suite of algorithms [Feldman *et al.*, 2013]. The main idea of LYDIA-NG is to perform multiple simulations for various hypothesized health states of the plant. The output of these multiple simulations is then processed and combined into single diagnostic output. LYDIA-NG has been successfully used for complex applications like space satellites [Feldman *et al.*, 2013]. However, when applied to online fault diagnosis of large dynamic systems, running all the hypothesized health states becomes a quite difficult and time consuming task.

Several approaches have been proposed in recent years to deal with the complexity issue. System decomposition methods, have been proposed to reduce the complexity in the fault diagnosis task [Bregon *et al.*, 2012] by generating smaller simulation submodels which can run in parallel and provide independent diagnosis decisions. The Possible Conflict, PC, approach [Pulido and Alonso-González, 2004], is an off-line dependency compilation technique from the DX community, which decomposes the global system model into minimal submodels, and performs on-line behavior estimation using simulation, dynamic bayesian networks, or state-based neural networks [Pulido *et al.*, 2012]. If a discrepancy is found, a set of fault candidates is generated by a minimal hitting-set algorithm of the triggered PCs. However, additional techniques must be used to refine the set of fault candidates.

The goal of this work consists of integrating PCs within the LYDIA-NG diagnosis framework. First, PCs will decompose the global simulation model into a set of smaller simulation submodels. Then, PCs will be used for efficient online fault detection and fault localization, providing a subset of fault candidates from the minimal hitting-set of the fault parameters linked to the set of equations in the PC. The subset of fault candidates is then used as input to LYDIA-NG, where simulations are run only for each one of the fault candidates, and its result is processed and combined to provide the diagnosis output. The approach has been tested by using a three-tank system case study.

The rest of the paper is organized as follows. Section 2 presents the basic definitions and running example used in this work. Section 3 briefly introduces LYDIA-NG and PCs. Section 4 presents our proposal to integrate PCs within the LYDIA-NG diagnosis framework. Section 5 describes the experimental results obtained for the three-tank system. Section 6 presents related work. And, finally, section 7 presents the discussion and conclusions.

2 Concepts and Definitions

In this section we present our basic definitions and a running example that we use to illustrate the significant concepts of this paper. Since both LYDIA-NG and PCs are model-based diagnosis approaches, we provide a set of definitions about models and faults that will allow us to explain later both techniques using the same framework.

2.1 Definitions

For the purpose of this work we focus our description on continuous systems, with only one nominal state, and whose behavior can be described as a set Σ of Ordinary Differential Equations (ODEs). The model of our system will be the basic system description to perform diagnosis:

*A. Bregon, B. Pulido, and C. Alonso's funding for this work was provided by the Spanish MCI TIN2009-11326 grant.

Definition 1 (Model)**.** The system model is defined as $M(\Sigma, U, Y, X, \Theta)$, where: Σ is a set of ODEs, defined over a collection of known and unknown variables: U is a set of inputs, Y a set of outputs, X a set of state and intermediate, i.e. unknown, variables, and Θ is the set of parameters[1].

Definition 2 (System Description, SD)**.** SD is made up of (M, H, σ, Π), where

- H is the health-vector defined by means of $h_i \mid 1 \leq i \leq k$ health variables, that allow us to characterize the set of states in the system, i.e. each $h_i \in H$ is a potential mode for the system, either nominal or faulty.
- σ is a mapping function: $\sigma(M, H_c) \rightarrow M_{Hc}(\Sigma_{Hc}, U_{Hc}, Y_{Hc}, X_{Hc}, \Theta_{Hc})$, that given the model, M, and the current health status, Hc, provides the model for behavior estimation for the current mode (or current system description M_{Hc}): $\Sigma_{Hc} \subseteq \Sigma$, $U_{Hc} \subseteq U$, $Y_{Hc} \subseteq Y$, $X_{Hc} \subseteq X$, and $\Theta_{Hc} \subseteq \Theta$).
- Π is a mapping function $\Pi(\theta_{cc}) \rightarrow \{Hc \mid Hc \subseteq H\}$ that, given a set of parameters, provides the set of health variables that relate to the set of model parameters: $\theta_{cc} \subseteq \Theta_{cc}$.

An implicit assumption in our modeling approach is that we can use the same set of equations for both the nominal behavior estimation and the faulty behavior estimation, just changing the value of θ_i.

In model-based diagnosis, the model of the system is used to compute a residual signal, which is used for fault detection and isolation purposes. A residual is computed as the difference between the observed behavior (obtained via sensor outputs y_i) and the expected behavior (estimated by the system model, $\hat{y}_i$), and it is formally defined as follows:

Definition 3 (Residual)**.** A residual is a real-valued measure $R(y_i, \hat{y}_i)$ of the difference between real and simulated system output at time t.

2.2 Running Example

In this paper, we use the three-tank system shown in Fig. 1 as the running example. The three tanks are denoted as T_1, T_2, and T_3. They all have the same area $A_1 = A_2 = A_3 = 3\,[\mathrm{m}^2]$. The experiments are performed assuming the gravity $g = 10$ and the liquid with density $\rho = 1$.

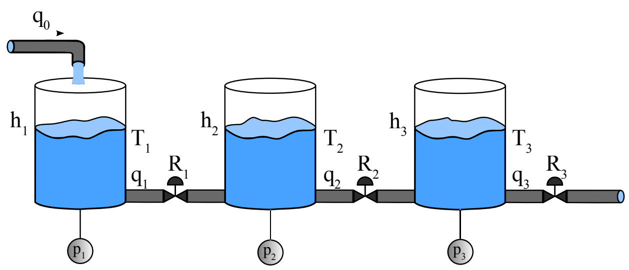

Figure 1: Diagram of the three-tank system.

Tank T_1 gets filled from a pipe q_0 with a constant flow of $1.5\,[\mathrm{m}^3/\mathrm{s}]$. It drains into T_2 via a pipe q_1. The liquid level is denoted as h_1. There is a pressure sensor p_1 connected to T_1 that measures the pressure in Pascals [Pa]. Starting from the Newton's (and Bernouli's) equations and manipulating them we derive the following Ordinary Differential Equation (ODE) that gives the level of the liquid in T_1:

$$\frac{dh_1}{dt} = \frac{q_0 - k_1\sqrt{h_1 - h_2}}{A_1} \tag{1}$$

In Eq. 1, the coefficient k_1 is used to model the area of the drainage hole and its friction factor. We emphasize the use of k_1 because, later, we will be "diagnosing" our system in term of changes in k_1. Consider a physical valve R_1 between T_1 and T_2 that constraints the flow between the two tanks. We can say that the valve changes proportionally the cross-sectional drainage area of q_1 and hence k_1. The diagnostic task will be to compute the true value of k_1, given p_1, and from k_1 we can compute the actual position of the valve R_1. The water levels of T_2 and T_3, denoted as h_2 and h_3 respectively, are given by:

$$\frac{dh_2}{dt} = \frac{k_1\sqrt{h_1 - h_2} - k_2\sqrt{h_2 - h_3}}{A_2}, \tag{2}$$

$$\frac{dh_3}{dt} = \frac{k_2\sqrt{h_2 - h_3} - k_3\sqrt{h_3}}{A_3}. \tag{3}$$

Values k_1, k_2, and k_3, are constant values with no physical meaning, and we have set them with a value of 0.75. Finally, we turn the water level into pressure:

$$p_i = \frac{g\, h_i\, A_i}{A_i} = g\, h_i \tag{4}$$

where i is the tank index ($i \in \{1, 2, 3\}$). To observe the behavior of the system we have an observational model, that allows us to know or read each value p_i. We use p_i^* to distinguish the measured variable from the model output p_i as $p_i^* = p_i$.

It is assumed that the initial water level in the three tanks is zero. Additionally, we make explicit the relation between the state variables, h_i in our example, and their derivatives, dh_i, as $h_i = \int dh_i \cdot dt$. These equations allow us to select an integral or differential approach for behavior simulation, depending on the selected causality. These equations make no influence in the diagnosis results, because they will have no θ_i, and consequently no health status.

3 Algorithms

This section presents the fundamental ideas of the LYDIA-NG diagnosis framework and the structural model decomposition approach with PCs.

3.1 LYDIA-NG

The basic idea of the LYDIA-NG diagnostic library is to perform multiple simulations for various hypothesized health states of the plant. The output of these simulations is then processed and combined into single diagnostic output.

The LYDIA-NG diagnostic library consists of the following building blocks: (i) Generator of Diagnostic Assumptions; (ii) Simulation Engine; (iii) Residual Analysis Engine; (iv) Candidate Selection Algorithm; (v) System State Estimation Algorithm. Detailed description of these blocks can be found in [Feldman *et al.*, 2013].

Algorithm 1 shows the top-level diagnostic process. The inputs to Algorithm 1 are a model and a scenario, and the

[1]Since we are dealing with fault diagnosis, in our model we are mainly interested in every parameter suitable to model faulty behavior.

result is a diagnosis. Algorithm 1 supports a large variety of simulation methods that may or may not use time as an independent variable. The only requirement toward the simulation engine is to predict a number of variables whose types can be mapped to LYDIA-NG and to be relatively fast.

Algorithm 1 Diagnosis framework

function DIAGNOSE(SCN) **returns** a diagnosis
  inputs: SCN, diagnostic scenario
  local variables: $\mathbf{h}$, FDI vector, health assignment
    $\mathbf{p}$, real vector, prediction
    Ω, a set of diagnostic candidates
    DIAG, diagnosis, result
  while $\mathbf{h} \leftarrow$ NEXTHEALTHASSIGNMENT() **do**
    $\mathbf{p} \leftarrow$ SIMULATE$(M, \gamma, \mathbf{h})$
    $r \leftarrow$ COMPUTERESIDUAL$(\mathbf{p}, \alpha)$
    $\Omega \leftarrow \Omega \cup \langle \mathbf{h}, r \rangle$
  end while
  DIAG $\leftarrow$ COMBINECANDIDATES(Ω)
  return DIAG
end function

The basic idea of Algorithm 1 is to simulate for various health assignments and to compare the predictions with the observed sensor data (i.e., telemetry). There are several important aspects of these algorithms that ultimately affect the diagnostic accuracy as measured by various performance metrics.

The first algorithmic property that determines many of the diagnostic performances is the order in which health-assignments are generated. In Algorithm 1 this is implemented in the NEXTHEALTHASSIGNMENT function. The latter subroutine also determines when to stop the search and should be properly parametrized depending on the model and the user requirements. In the standard LYDIA-NG diagnostic library we provide the breadth-first search (BFS), the depth-first search (DFS), and the backwards greedy stochastic search (BGSS) diagnostic search policies.

Each simulation produces a set of predicted values for a given health-assignment. The second important property of Algorithm 1 is the comparison and ordering of the diagnostic candidates. This is done by mapping the predicted and observed variables into a single real-number, called a *residual*.

Residual generation functions in LYDIA-NG bear resemblance to loss functions in decision theory. For example, residuals may be squared or absolute residuals [Feldman *et al.*, 2013]. A disadvantage of the squared residuals function is that it adds a lot weight to outliers.

3.2 Consistency-based diagnosis with PCs

In this section we present the fundamental ideas of Consistency-based Diagnosis and Possible Conflicts.

Consistency-based Diagnosis

Consistency Based Diagnosis (CBD) performs fault detection and fault isolation using only models of correct behavior in a *two stage process*. First, we identify if there exists a discrepancy between the observed behavior and the expected behavior, thus defining a discrepancy in terms of a residual. Corresponding to each residual, or discrepancy, is a conflict [Reiter, 1987]. Hence, fault detection consists of computing every conflict.

The second step is fault isolation, which consists of computing the minimal hitting sets of the conflicts, since they characterize the whole set of minimal diagnoses [Reiter, 1987]. Intuitively, a conflict is a set of components that cannot behave properly simultaneously, given the system description and current observations of abnormal behavior. In this work we use the Possible Conflicts (PCs) approach to avoid the on-line computation of conflicts and speed up overall fault isolation. PCs are designed to compute off-line those subsystems capable of becoming conflicts online.

For consistency-based diagnosis using PCs, we only use $\sigma(M, Hn)$ with Hn corresponding to a nominal mode. Since we are dealing with a continuous system working in one nominal mode, we can compute offline the set of PCs for $M_{Hn}(\Sigma_{Hn}, U_{Hn}, Y_{Hn}, X_{Hn}, \Theta_{Hn})$, as will be described later. The output of the consistency-based diagnosis using PCs is a set of fault candidates C defined in the lattice provided by Θ^*.

Model decomposition with PCs

The Possible Conflicts (PCs) approach [Pulido and Alonso-González, 2004] is a model decomposition method that finds (off-line) every subset of equations capable of generating conflicts. PCs provide the structural and causal model of a subsystem with minimal redundancy. The set of equations in a PC can be used to simulate the correct behavior of the subsystem. Hence, PCs can be used in CBD of dynamic systems [Pulido *et al.*, 2001]. For the sake of self-containment, we summarize here the proposal for PCs computation given in [Pulido and Alonso-González, 2004].

To compute PCs, we need the structural model of the system under study, which can be obtained from the set of equations in the system description, once we select a given working mode, tailored for our new problem formulation, instead of the original process which was suitable for system descriptions provided as hypergraphs [Pulido and Alonso-González, 2004]. We will illustrate the process using the three-tank system in Fig. 1, and the set of equations in its model as described in Section 2.2.

We need an abstraction of our model description $SD = (M, H, \sigma, \Pi)$. Let's assume we compute the set of PCs for a given nominal mode characterized by H_n. Using $\sigma(M, H_n)$, we obtain $M_{H_n} = (\Sigma_{H_n}, U_{H_n}, Y_{H_n}, X_{H_n}, \Theta_{H_n})$. For the structural model, we only need the information about the measured and unknown variables in each model equation. Thus each equation $\sigma_i \in \Sigma_{H_n}$ will provide one structural constraint $\sigma_i \rightarrow (S_i,\ X_i)$, where S_i accounts for the measured variables from U_{H_n}, Y_{H_n} in σ_i, and X_i accounts for the unknown (state or intermediate variables in σ_i).

For the three-tank system the structural model is made up of the following constraints:

Constraint	*Sensors*	*Unknowns*
c_1	$\{q_0\}$	$\{d_{h1}, h_1, h_2\}$
c_2	$\{\}$	$\{d_{h2}, h_1, h_2, h_3\}$
c_3	$\{\}$	$\{d_{h3}, h_2, h_3\}$
c_4	$\{\}$	$\{p_1, h_1\}$
c_5	$\{\}$	$\{p_2, h_2\}$
c_6	$\{\}$	$\{p_3, h_3\}$
c_7	$\{p_1^*\}$	$\{p_1\}$
c_8	$\{p_2^*\}$	$\{p_2\}$
c_9	$\{p_3^*\}$	$\{p_3\}$
c_{10}	$\{\}$	$\{h_1, d_{h1}\}$
c_{11}	$\{\}$	$\{h_2, d_{h2}\}$
c_{12}	$\{\}$	$\{h_3, d_{h3}\}$

where constraints c_1 to c_3 are related to equations (1) to (3); constraints c_4 to c_6 are related to the equation (4) for each one of the tanks; constraints c_7 to c_9 make explicit the diagnosis observational model, relating the output variable p_i and its associated sensor p_i^*; and constraints c_{10} to c_{12} make explicit the dynamic in the system: relation between the state variable and its derivative.

The first step in PC computation is to look for the complete set of minimally redundant subsets of equations, which we call the *Minimal Evaluation Chains* (MECs). A MEC represents a strictly overdetermined[2] set of equations that can potentially be solved using local propagation (elimination method): each MEC will have n constraints and $n-1$ unknowns. A summary of the algorithms used to compute MECs in a system can be found in [Pulido and Alonso-González, 2004]. The set of MECs in the system in Fig. 1 is:

- $mec_1 = \{c_7, c_4, c_{10}, c_1, c_5, c_8\}$
- $mec_2 = \{c_8, c_5, c_{11}, c_2, c_4, c_6, c_7, c_9\}$
- $mec_3 = \{c_9, c_6, c_{12}, c_3, c_5, c_8\}$

We need to know the different ways an equation can be solved, because we can deal with non-linear models. These ways are usually called the set of possible causal assignments for the variables in an equation. We assume that the set of possible causal assignments is known for the system model, and we build the complete set of valid causal assignments for the set of MECs, using exhaustive search [Pulido and Alonso-González, 2004]. We call each valid causal assignment *Minimal Evaluation Model* (MEM).

For the three-tank system, we assume that the causality is given by the expression in equations (1) to (6), except for the observational model (in this case we allow solving constraints ec_7 to ec_9 in both directions because we need to convert some system measurements Y in MEM inputs, U_{pc}). The set of MEMs for the three-tank system and their discrepancy nodes are shown in Table 1.

Table 1: MEMs for the three-tank system and their discrepancy nodes.

MEM	Discrepancy	Parameters
$\{c_7, c_4, c_{10}, c_1, c_5, c_8\}$	p_1^*	k_1, A_1
$\{c_8, c_5, c_{11}, c_2, c_4, c_6, c_7, c_9\}$	p_2^*	k_1, k_2, A_2
$\{c_9, c_6, c_{12}, c_3, c_5, c_8\}$	p_3^*	k_2, k_3, A_3

Fault detection and isolation using PCs

In the MEM there is a special node called *discrepancy* node (representing the only variable that is estimated by two different ways). Therefore, that node is the potential source of a residual, or discrepancy, using only the values of measured variables as inputs, and the past value of state-variables.

In CBD [Reiter, 1987; de Kleer and Williams, 1987] a conflict arises given a discrepancy between observed and predicted values for a variable. Under fault conditions, conflicts are observed when the model described by a MEM is evaluated with available observations and produce a discrepancy, because the model equations and the input/measured values are inconsistent [Reiter, 1987; de Kleer and Williams, 1987]. This notion of possible discrepancy generation leads to the definition of *Possible Conflict*:

Definition 4 (Possible Conflict)**.** The set of constraints in a MEC that give rise to at least one MEM.

Every MEC in the three-tank system has one MEM. Hence, we have three PCs. Each MEM is the computational model for a PC, and each equation in a MEM contains zero or more parameters that can be the source of potential faults (θ_{cc} in our model description). The set of parameters related to each PC is also shown in the fourth column in Table 1. Given a non-zero residual, we then isolate the fault parameters involved in the *pc* structural model: Θ_{pc}. This information is the basis for the integration of Consistency-based diagnosis of dynamic systems with Possible Conflicts and LYDIA-NG.

4 On-line Fault diagnosis with LYDIA-NG and PCs

In CBD, diagnosis must discriminate among 2^N behavioral mode assignments when just correct, $ok(\cdot)$, and incorrect modes, $\neg ok(\cdot)$, are present for N components. When B behavioral models are allowed, diagnosis must discriminate among B^N mode assignments. This is the problem faced by any model-based diagnosis proposal which attempts fault identification [Dressler, 1996], and it is also present in LYDIA-NG. In this section, we present an integration proposal, where the system model is partitioned using PCs. As explained in Section 2, the output of the consistency-based diagnosis using PCs is a set of fault candidates C defined in the lattice provided by Θ^*. Then, this set of diagnosis candidates is used as input to LYDIA-NG, thus reducing the number of health state simulations that needs to be considered by LYDIA-NG.

In our integration proposal, the simulation model for each PC uses some of the system measurements as input, and provides an estimation for exactly one variable (the potential discrepancy). Then, an executable model, SD_{pc_i} for each pc_i, is built. This executable model can be a simulation model, a state observer, or even a neural network [Pulido *et al.*, 2012]. Summarizing, the integration of LYDIA-NG and CBD with PCs is possible given the set of candidates, C: each candidate C_i is a subset of Θ_{pc}. Then invoking $\Pi(C_i)$, LYDIA-NG can obtain the set of health variables, Hc related to C_i, and use them as input for its search. Given the current implementation of LYDIA-NG, we can obtain the system description (system model) imposed by $Hc : \sigma(M, Hc)$, which is enough to characterize the current model and perform simulation of the Hc health status.

Algorithm 2 shows the algorithm for our integrated diagnosis framework. Y_{pc_i} denotes the set of input observations available for the executable model of a PC, SD_{pc_i}; and $\hat{Y}_{pc_i}$ represents the set of predictions obtained from SD_{pc_i}. The function OBTAINOBSERVATIONS obtains from the diagnostic scenario the observations which have to be used as input for each PC. Function ESTIMATEBEHAVIOR provides an estimation of a measured variable by using the executable model of each PC (either a simulation model, a state observer model, or a neural network).

For the detection part, to determine significant deviations from the PC residuals (PC residuals are computed by using an absolute residual function). We use the Z-test for robust fault detection using a set of sliding windows as detailed

[2]A redundant set of equations would be an Evaluation Chain. Since we are interested only on minimal conflicts, we just focus on the set of MECs that are by definition minimally overdetermined.

in [Daigle *et al.*, 2010]. A small window, N_2, is used to estimate the current mean of the residual signal, μ_r. The variance of the nominal residual signal is computed using a large window N_1 preceding N_2, by a buffer N_{delay}, which ensures that N_1 does not contain any samples after fault occurrence. The variance and the confidence level determined by the user are then used to dynamically compute the detection thresholds ϵ_r^- and ϵ_r^+.

Algorithm 2 Integrated PCs and LYDIA-NG diagnosis approach.

function PCS-LYDIA-DIAGNOSIS(SCN) **returns** a diagnosis
  inputs: SCN, diagnostic scenario
  local variables: Y_{pc_i}, set of input observations
    $\hat{Y}_{pc_i}$, estimation from the PC
    Θ_{pc_i}, fault parameters involved in the PC
    $\mathbf{h}$, FDI vector, health assignment
    $\mathbf{p}$, real vector, prediction
    Ω, a set of diagnostic candidates
    DIAG, diagnosis, result
  repeat
    $Y_{pc_i} \leftarrow$ OBTAINOBSERVATIONS(SCN)
    $\hat{Y}_{pc_i} \leftarrow$ ESTIMATEBEHAVIOR(SD_{pc_i}, Y_{pc_i})
    $r_{pc_i} \leftarrow$ COMPUTERESIDUALPC($\hat{Y}_{pc_i}, Y_{pc_i}$)
    if Z-TEST(r_{pc_i}) $< \epsilon_r^-$ or Z-TEST(r_{pc_i}) $> \epsilon_r^+$ **then**
      Θ_{pc_i} = confirm pc_i as a real conflict
      $C \leftarrow \text{MHS}(C, \Theta_{pc_i})$
    end if
  until Every pc_i is activated or time elapsed or a unique fault candidate has been isolated
  while $\mathbf{h} \leftarrow$ NEXTHEALTHASSIGNMENT(Π, C) **do**
    $\mathbf{p} \leftarrow$ SIMULATE($M, \gamma, \mathbf{h}$)
    $r \leftarrow$ COMPUTERESIDUALLYDIA($\mathbf{p}, \alpha$)
    $\Omega \leftarrow \Omega \cup \langle \mathbf{h}, r \rangle$
  end while
  DIAG $\leftarrow$ COMBINECANDIDATES(Ω)
  return DIAG
end function

Once the initial set of fault candidates has been isolated, the LYDIA-NG part of the algorithm is run (as shown in algorithm 2). The algorithm takes the set of isolated fault candidates as input, and the NEXTHEALTHASSIGNMENT function only considers the health assignments related to the fault candidates. In this version of the integrated framework, the global system model is used as the simulation model, instead of the PC submodels, thus providing a more direct way to integrate both approaches. In future versions, the PC submodels will also be used as the simulation model in LYDIA-NG, thus providing faster simulation results.

5 Results

In this section we show some diagnosis results for our integrated framework. We first discuss the nominal situation, then, we present an on-line fault diagnosis scenario for a particular fault in the three-tank system and discuss the results obtained.

5.1 Nominal Scenario

For the nominal scenario, none of the three PCs found for the system is triggered. The advantage of including PCs within the LYDIA-NG framework is evident in this case. Since none of the PCs is triggered, LYDIA-NG is not run, thus avoiding the time-consuming simulations for the different health states when no actual fault has occurred in the system.

5.2 On-line fault diagnosis

This section briefly describes how LYDIA-NG runs with and without the use of PCs. The first phase is residual analysis, where LYDIA-NG runs a set of simulations such that a residual is computed for each simulation. Because LYDIA-NG uses real-value health variables, the space of potential diagnostic assumptions, and the corresponding set of simulations, is enormous, and infinite in the worst case. The heuristics used for generation of diagnostic assumptions are critical to the success and efficiency of LYDIA-NG.

LYDIA-NG ranks the residual outputs, discarding those candidates whose residual value is larger than the residual of the "all nominal" candidate. The remaining candidates are assigned probabilities of occurrence, using a method described in [Feldman *et al.*, 2013]. The fault isolation process assigns probabilities of failure to system components, and these are reported as ranked diagnoses.

In the following we compare the results for running LYDIA-NG with and without PCs. Without PCs, LYDIA-NG uses the global system model described earlier; with PCs (i.e., using Algorithm 2), the generation of diagnostic assumptions is governed by the PC-based algorithm.

For a diagnosis scenario with a 40% blockage fault in valve R_1 occurring at time 100 s, our results are as follows.

Non-PC-based Approach: LYDIA-NG computes residuals based on the difference between the pressures. The non-zero residual at time 104 s creates a set of simulations in which LYDIA-NG analyzes several valve %-blockage cases for R_1, R_2 and R_3. LYDIA-NG estimates the valve positions by "guessing" the *true* valve positions and computes the health probability by subtracting the commanded valve position from the estimated one. LYDIA-NG is able to isolate the most-likely fault as (R_1, 40%).

PC-based Approach: When computing diagnoses for this fault, at time 104 s, an increase in the residual of PC_2 is detected, and consequently k_1, k_2, and A_2 are selected as the initial set of fault candidates. At the next time step, at time 105 s, PC_1 is triggered, thus selecting k_1 and A_1 as possible fault candidates. A minimal hitting set algorithm is run, determining that the only single fault candidate in the system is k_1. At this point, the fault identification for k_1 is triggered by using LYDIA-NG.

Running this diagnosis scenario with a (trivial) input of R_1 (as derived from the candidate k_1), as opposed to R_1, R_2 and R_3, results in an 80× speedup of LYDIA-NG as compared to the non-PC approach. This is a result of reducing the diagnosis assumption space.

6 Related work

LYDIA-NG belongs to a class of MBD methods that use continuous-valued models and sensor data, and use entropy based methods for test selection to disambiguate diagnoses. It is a generalization of LYDIA-NG, which used discrete-value models.

In terms of diagnostics solvers, LYDIA-NG is related to the HyDE (Hybrid Diagnosis Engine) solver [Narasimhan and Brownston, 2007]. Another solver, FACT [Daigle *et al.*, 2010], can also use continuous-valued models and sensor data, but requires that the model be represented as a hybrid bond graph. Given an anomaly, FACT first uses an observer-based approach (adopted from the FDI community) with statistical techniques for robust fault detection.

Recent works have demonstrated the similarities between model-based diagnosis approaches from the DX and the FDI communities [Cordier *et al.*, 2004]. In such framework, it has been demonstrated the equivalence of several structural model decomposition techniques such as PCs, minimal ARRs and Minimally Structurally Overdetermined sets [Armengol *et al.*, 2009]. As a consequence, the proposal in this work can be easily extended to other structural methods.

Using CBD we need to generate the set of candidates C and wait for every PC to be confirmed. An FDI approach would use exoneration using the structural information in the set of PCs. In CBD we wait for additional observations in order to reject modes that are not consistent with available information. Combining our results with LYDIA-NG provides an additional boost for candidate discrimination by including fault models through health variables.

7 Conclusions

This work has presented an integrated framework for on line fault detection, isolation and identification of dynamic systems.

Two different approaches have been integrated: The LYDIA-NG suite of diagnosis algorithms and the PCs framework for on-line CBD. LYDIA-NG is a simulation based diagnosis system that filters out diagnosis candidates discarding those of them that generates residuals larger than the *all-nominal* assumption, i.e., fault free and nominal system configuration. Although the system incorporates important facilities, such as diagnostic test generation based on entropy measure, its main drawback is the lack of focus for the initial set of candidates, which may be large, and the cost of simulating the complete system for every considered candidate. On the contrary, the set of PCs identifies minimal computational subsystems that decompose the complete system and that can be simulated independently. PCs are based on Reiter's theory of diagnosis from first principles and are able to generate fault isolation candidates from model of correct behavior without hypothesizing an initial set of candidates. Hence, using consistency-based diagnosis with PCs candidate generation is rather efficient, although additional techniques are required to further refine fault candidates for fault isolation and identification. They also lack some of the facilities incorporated in LYDIA-NG, like generation of diagnostic tests.

Our three-tank system running example shows the potential of this approach. First, when the system is fault free, no PC becomes a real conflict and no candidate is generated. This avoids running LYDIA-NG for fault detection, which is performed by the PCs approach, thus potentially providing a significant saving on computing time, depending on the size of the complete system and on the number and overlapping degree of the PCs. Second, when a fault is detected, PCs may generate a low number of fault candidates, depending on the number of PCs and its overlapping degree but also on the real faulty parameter, thus providing an automatic focus for LYDIA-NG fault candidate search.

References

[Armengol *et al.*, 2009] J. Armengol, A. Bregon, T. Escobet, E. Gelso, M. Krysander, M. Nyberg, X. Olive, B. Pulido, and L. Travé-Massuyès. Minimal Structurally Overdetermined sets for residual generation: A comparison of alternative approaches. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS09*, pages 1480–1485, Barcelona, Spain, 2009.

[Bregon *et al.*, 2012] A. Bregon, G. Biswas, and B. Pulido. A Decomposition Method for Nonlinear Parameter Estimation in TRANSCEND. *IEEE Trans. Syst. Man. Cy. Part A*, 42(3):751–763, 2012.

[Cordier *et al.*, 2004] M.O. Cordier, P. Dague, F. Lévy, J. Montmain, M. Staroswiecki, and L. Travé-Massuyès. Conflicts versus Analytical Redundancy Relations: a comparative analysis of the Model-based Diagnosis approach from the Artificial Intelligence and Automatic Control perspectives. *IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics*, 34(5):2163–2177, 2004.

[Daigle *et al.*, 2010] M. Daigle, I. Roychoudhury, G. Biswas, X. Koutsoukos, A. Patterson-Hine, and S. Poll. A comprehensive diagnosis methodology for complex hybrid systems: A case study on spacecraft power distribution systems. *IEEE Transactions of Systems, Man, and Cybernetics, Part A*, 4(5):917–931, September 2010.

[de Kleer and Williams, 1987] J. de Kleer and B. C. Williams. Diagnosing multiple faults. *Artificial Intelligente*, 32:97–130, 1987.

[Dressler, 1996] O. Dressler. On-line diagnosis and monitoring of dynamic systems based on qualitative models and dependency-recording diagnosis engines. In *Proceedings of the Twelfth European Conference on Artificial Intelligence, ECAI-96*, pages 461–465, 1996.

[Feldman *et al.*, 2013] Alexander Feldman, Helena Vicente de Castro, Arjan van Gemund, and Gregory Provan. Model-based diagnostic decision-support system for satellites. In *Aerospace Conference, 2013 IEEE*, pages 1–14. IEEE, 2013.

[Isermann, 2006] R. Isermann. *Fault-Diagnosis Systems. An Introduction from Fault Detection to Fault Tolerance*. Springer, 2006.

[Narasimhan and Brownston, 2007] S. Narasimhan and L. Brownston. Hyde–a general framework for stochastic and hybrid model-based diagnosis. In *Proc. 18th International Workshop on Principles of Diagnosis (DX07), Nashville, USA*, pages 162–169. Citeseer, 2007.

[Pulido and Alonso-González, 2004] B. Pulido and C. Alonso-González. Possible Conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics*, 34(5):2192–2206, Octubre 2004.

[Pulido *et al.*, 2001] B. Pulido, C. Alonso, and F. Acebes. Lessons learned from diagnosing dynamic systems using possible conflicts and quantitative models. In *Engineering of Intelligent Systems. XIV Conf. IEA/AIE-2001*, volume 2070 of *LNAI*, pages 135–144, Budapest, Hungary, 2001.

[Pulido *et al.*, 2012] B. Pulido, J.M. Zamarreño, A. Merino, and A. Bregon. Using structural decomposition methods to design gray-box models for fault diagnosis of complex systems: a beet sugar factory case study. In A. Bregon and A. Saxena, editors, *Procs. of the First European Conference of the Prognostics and Health Management Society*, pages 225–238, Dresden, Germany, July 2012. www.phmsociety.org.

[Reiter, 1987] R. Reiter. A Theory of Diagnosis from First Principles. *Artificial Intelligence*, 32:57–95, 1987.